

REMARKS

This is in response to the Office Action mailed on August 26, 2004. Claims 1, 7, 14, 22, 31-42, and 48 are amended. Claims 1-51 are now pending in this application. These amendments are consistent with the original filed specification and Applicants do not believe that any new matter has been introduced. Accordingly, entry of the amendments is appropriate.

Amendments to Specification

The specification has been amended to identify the application entitled "Caching of Dynamic Arrays" with its serial number, publication number and filing date. Applicants note that the identified docket number 303.743US1 was incorrect, and that the correct docket is 303.739US1. Docket 303.743US1 is the docket of the present application. Applicant respectfully asserts that no new matter is added, and requests the Examiner to consider and approve the amendment in the next action.

§101 Rejection of the Claims

Claims 1-41 were rejected under 35 USC § 101 as being directed to non-statutory subject matter. The Examiner indicated that the method claims were rejected because it did not limit the preamble to a media and thus the Examiner concluded the methods could be abstract ideas. Although, Applicants disagree with the Examiner's interpretation of section 101; Applicants have amended the independent claims 1, 7, 14, 22, 34, and 38 to positively recite within the preambles that the methods are "implemented in a computer-readable medium." Applicants believe this is what the Examiner asked for and complies with what the Examiner believes is necessary. Therefore, Applicant respectfully requests withdrawal of the 101 rejections.

§103 Rejection of the Claims

Claims 1-51 were rejected under 35 USC § 103(a) as being unpatentable over Tremblay et al. (U.S. 6,014,723) in view of admitted prior art. It is of course fundamental that in order to sustain an obviousness rejection that each and every step or element of the rejected claims must be taught or suggested in the proposed reference or proposed combination of references.

At the outset, Applicants respectfully disagree with the Examiner's interpretation of the Applicants' statements included in the Background of the Invention Section of the original filed specification. That is, Applicants do not concur that the Examiner's interpretation of these statements are considered admitted prior art. Accordingly, Applicants reserve the right to contest these interpretations at a latter point in time.

Tremblay relates to techniques that provide for an improved virtual machine translator. An aspect of achieving an improved virtual machine translator is stack processing and array bounds checking. The aspect of Tremblay that the Examiner has recited as being relevant to Applicants' claims is the Tremblay's array bound checking technique. In Tremblay, array bound checking is achieved by recording minimum and maximum bounds for identified arrays. During execution, if a particular array identifier is being processed then the Tremblay processing compares dynamic accesses to the previously recorded minimum and maximum bounds for that identified array. If an out-of-bounds condition occurs then Tremblay executes a pre-defined exception routine.

However, Applicants are unable to find, among other things in Tremblay, a set of user-defined instructions can dynamically overlay an existing array with a *user-defined set of statements* to dynamically define a dimensional overlay on that *existing array*. *Emphasis Added*. Additionally, the exception routine of Tremblay does not appear *to be defined and identified* by a user-defined set of statements. *Emphasis added*. The exception routing is predefined within the logic of Tremblay and the user does not have access to its processing logic. A user-defined set of statements cannot be used in Tremblay to dynamically define an overlay of an existing array. Furthermore, the user has no control in Tremblay to define, identify, and associate a user-defined array boundary policy with a user-defined overlaid array.

More specifically, and with respect to Applicants' amended independent claims 1, 7, 14, and 42, Applicants' amended claims further clarify that the dynamic overlay to a dynamic definition identified by statements is associated with a *user-defined* software program. *Emphasis Added*. Applicants are unable to find a showing or fair suggestion of this in Tremblay. Tremblay requires and needs to know the minimum and maximum values for an array, and an array definition (*single statement*) that must include the minimum and maximum values for a new array. *Emphasis Added*. Tremblay, col. 25, lines 14-24. A block of statements is not

dynamically used within a user-defined program in Tremblay to dynamically define an overlay of an *existing array*; rather in Tremblay a user program uses *a single statement* that includes the minimum and maximum values for a *new array*. *Emphasis Added.*

Tremblay is not capable of permitting a user to dynamically overlay an existing array with a block of statements that when evaluated defines a new dimensional array from an existing array. In order for Tremblay to support or teach such a situation; Tremblay would have had to have significant programming language enhancements to the user-defined programming language. This is described and detailed in Applicants original filed specification. Tremblay does not alter or provide enhancements to an existing programming language; rather Tremblay provides a run-time mechanism for a programming language that is translated into machine code for processing which provides for better real-time detection of references that occur outside the bounds of an array during execution of that program.

Therefore, with respect to Applicants' independent claims 1, 7, 14, and 42, Applicants respectfully request withdrawal of the rejection.

With respect to Applicants' amended independent claims 22, 34, 38, and 48; Applicants are unable to find a showing or fair suggestion in Tremblay of a user-defined set of statements can define and associate an array boundary policy. The present subject matter provides enhancements to user-defined programming languages that permit a developer to automatically associate within the developer's programming language a specific and user-defined (developer-defined) policy to process when an out-of-bounds condition is detected at run-time for the user-defined (developer-defined) program. Applicants are unable to find a showing or fair suggestion in Tremblay of this feature. In Tremblay, a pre-defined exception routine is automatically processed on behalf of a user when an out-of-bounds condition is detected. There is no ability at all in Tremblay that would permit a user to add logic within a user-defined program that would alter this exception processing for different user-defined arrays.

Thus, Tremblay fails to teach the ability to associate user-defined array boundary policies within a user-defined program as is positively recited in Applicants' amended claims 22, 34, 38, and 48. Therefore, the rejections with respect to these claims cannot be sustained and should be withdrawn.

Conclusion

Applicants respectfully submit that the claims are in condition for allowance, and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicants' attorney at (612) 373-6960 to facilitate prosecution of this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

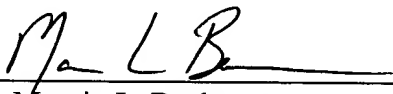
Respectfully submitted,

KEITH R. SLAVIN ET AL.

By their Representatives,

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938
Minneapolis, MN 55402
(612) 373-6960

Date 11-23-04

By 
Marvin L. Beekman
Reg. No. 38,377

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: MS Amendment, Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 23 day of November, 2004.

KACIA LEE
Name

Kacia Lee
Signature